# AJAX : JQUERY

LCProd.net
Grandadam Emmanuel

2

Le modèle Client/Serveur
Utiliser javascript
Pourquoi un framework ?
JQuery / JQuery UI
Débugger et tester le code

# INTRODUCTION / WEB 2.0

LCProd.net
Grandadam Emmanuel

# CLIENT/SERVEUR

## Modèle Web

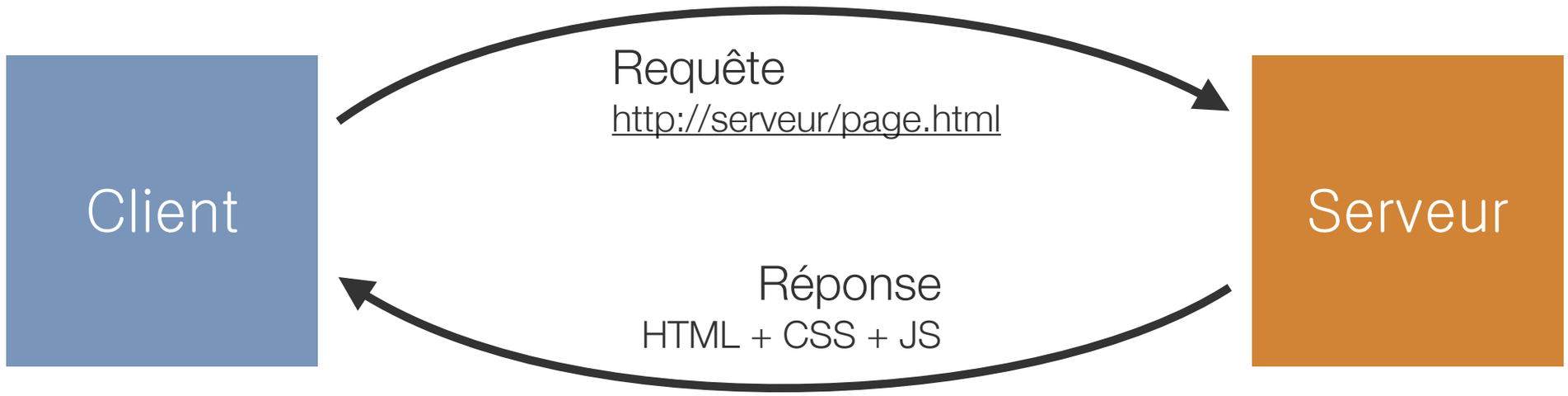
- Questions/Réponses

  *Requête http://....*

- Retour normalisé :

  *XHTML/HTML5/CSS*

- Le javascript est dans la page et est exécuter par le client !



# UTILISER JAVASCRIPT

**«JavaScript est un langage interprété par le navigateur. Le JavaScript est un langage « client », c'est-à-dire exécuté chez l'utilisateur lorsque la page Web est chargée. Il a pour but de dynamiser les sites Internet.»**

## Les navigateurs :

- Chaque navigateur interprète à sa façon Javascript et possède ses propres objets

*-> Difficilement compatible entre les différents navigateurs.*

*-> Test du W3C : ACID 1/2/3*

**http://en.wikipedia.org/wiki/Acid3**

# UTILISER JAVASCRIPT



## A ne pas oublier :

- Javascript n'est pas sécurisé !
  *Ne pas faire confiance à une donnée provenant du client*

- On doit pouvoir sans passer...
  *Lynx est un navigateur sans javascript*

- On ne peux garantir le résultat...
  *La page doit fonctionner sans Javascript*

# POURQUOI UN FRAMEWORK ?

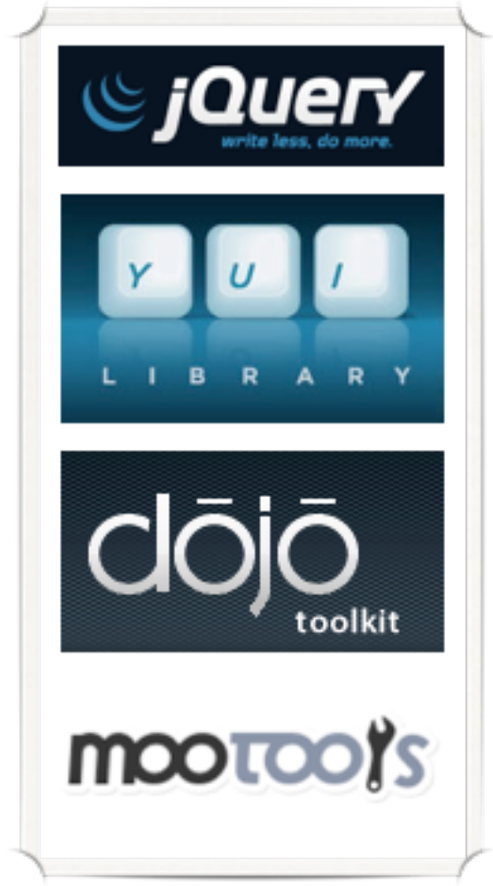


## 5 raisons (et bien plus)

- S'abstraire des problèmes de compatibilité des navigateurs
- Code normalisé
- Méthodes simplifiée
- Gestion de plug-ins
- Forte communauté

## et aussi...

- Documentation disponible
- Actualisé
- Evolutif
- ...



# JQUERY / JQUERY UI

## JQuery

**http://jquery.com/**


- Noyau de base qui gère :
  *Les sélecteurs,*
  *Les événements,*
  *Ajax*

## JQuery UI

**http://jqueryui.com/**


- Fonctionnalités pour gérer l'Interface Utilisateur
- Demande le noyau JQuery pour fonctionner
  *Effets de déplacement (drag),*
  *Effets pour re-dimensionner les objets,*
  *Outils pour les interactions de l'utilisateur (dialogues...)*

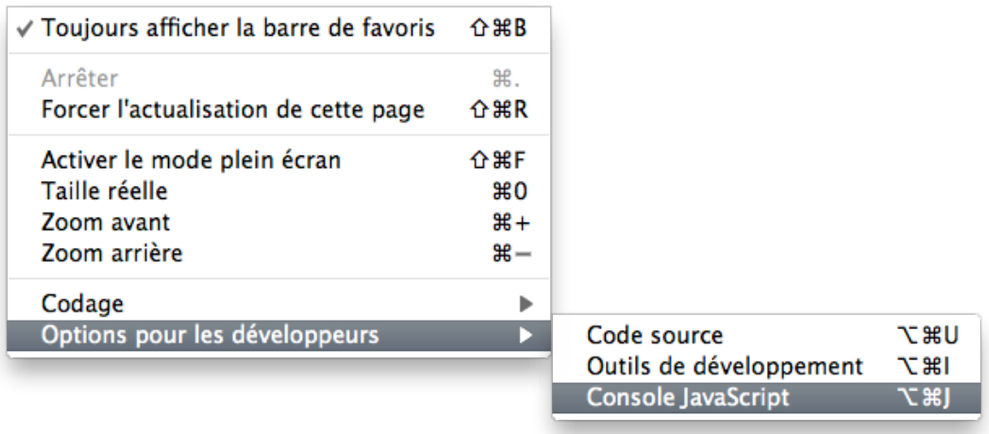
# DEBUGGER ET TESTER



## Les navigateurs 'modernes' intègre un outil de debuggage

- Pour Chrome/Safari

  *Menu 'présentation' ->*
  *'Options pour les développeurs' ->*
  *Console JavaScript*




- Pour Firefox

  *Utiliser le module 'Web Developer'*


**https://addons.mozilla.org/fr/firefox/addon/web-developer/**

9

DOM
Scripts
Chargement dynamique
XMLHTTPRequest

# COMPRENDRE LE PRINCIPE

LCProd.net
Grandadam Emmanuel

# DOM



## Document Object Model

- Modèle objets de la page fabriqué par le navigateur
- Permet la manipulation du contenu au format XML
- C'est une représentation logique de la page HTML

*Les transformations appliquées au DOM sont renvoyées à l'affichage*

```
└HTML
  └HEAD
    └TITLE
      └#text: An HTML Document
    BODY
    └H1
      └#text: Example
      P
      └#text: This is an example HTML document.
```

# SCRIPTS

## Deux emplacements :

- Dans la page HTML

*XHTML*

```
<script type="text/javascript"> ... </script>
```

*HTML5*

```
<script>...</script>
```

- Dans un document externe .js

*XHTML*

```
<script type="text/javascript" src="fichier.js"></script>
```

*HTML5*

```
<script src="fichier.js"></script>
```

# SCRIPTS

## Intégrer la bibliothèque JQuery :

*Depuis le site Officiel http://jquery.com*

```
<script src="jquery.js"></script>
```

*Utilisé depuis les API Google http://code.google.com/intl/fr/apis/libraries/*

```
<script src="http://www.google.com/jsapi"></script>
<script>google.load("jquery", "1.5.2");</script>
```

## Attendre la chargement complet du DOM pour commencer à exécuter le code javascript

- Javascript :

```
window.onload =  function(){ ... };
```

*JQuery :*

```
$(document).ready(function() { ... });
$(function(){ ... });
```

# CHARGEMENT DYNAMIQUE

## Dynamic Script Loading

- Javascript permet le chargement dynamique de scripts externes...

*Exemple 1*

```
document.write("<script src='fichier.js'><\/script>");
```

*Exemple 2*

```
var monScript  = document.createElement("script");
monScript.src  = "fichier.js";
monScript.type = "text/javascript";
document.body.appendChild(monScript);
```

# XMLHTTPREQUEST



## Objet ActiveX ou Javascript:

*Permet d'obtenir des données au format HTML, JSON ou XML à l'aide d'une requête HTTP en arrière plan.*

- Requête synchrone

  *La page est 'bloquée' pendant la requête jusqu'au chargement des données reçues.*

  **Exemple : La page se recharge totalement pour afficher la suite.**

- Requête asynchrone

  *La page reste 'active' pendant que la requête s'exécute en arrière plan.*

  **Exemple : La page ne 'bouge' pas, le résultat de la requête est intégré à une partie de la page.**

15

Syntaxe du langage
Les variables
Structures conditionnelles
Structures de boucles
Structures de dialogues
Les objets
Les fonctions

# CODER EN JAVASCRIPT

LCProd.net
Grandadam Emmanuel

# SYNTAXE DU LANGAGE

## Les opérateurs

- Nombres et textes

```
var1 = 1;              // nombre
var2 = '1';            // texte
var3 ++;               // incrément
```

- Logique

```
var1 && var2;          // ET logique
var1 || var2;          // OU logique
```

- Comparaison

```
var1 == var2;          // égal
var1 != var2;          // différent
```

# LES VARIABLES

## Utilisation de variables

- Déclaration

```
var mavariable;
var mavariable = 1;

var mavariable = 1,
secondevar = "bonjour",
troisemevar = function();
```

- Portée des variables

*Une variable déclarée dans une fonction n'est accessible que dans cette fonction.*
*Une variable déclarée dans l'entête du script est accessible partout.*

# LES VARIABLES

## Utilisation des tableaux (array)

- Déclaration

```
var monTableau = [ 'Tintin', 'Milou' ];
```

- Fonction sur les tableaux

*Méthode push() Ajoute un élément*

```
monTableau.push('Haddock');
```

*Méthode join() Réunis les éléments*

```
montableau.join('');          // "TintinMilouHaddock"
```

*...*

# CONDITIONNELLES



## Syntaxe

- If ... Else

```
if (var == 1){
    ...
}else{
    ...
}
```

- Switch

```
switch(variable){
   case 'a':
      ...
      break;
   case 'b':
      ...
      break;
}
```

# STRUCTURES DE BOUCLES



## Syntaxe

- for

```
for (var i = 1; i < 100; i++){ ... }
```

- while

*Pas réalisée si i est supérieur à 100*

```
while(i < 100){
    i++;
}
```

- do ... while

*Réalisée au moins une fois*

```
do{
    ...
}while (false);
```

# STRUCTURES DE DIALOGUES

## Interactions avec l'utilisateur

- Méthode alert()

```
alert('Bonjour');
```

- Méthode confirm();

```
confirm('Question ?');                    // true ou false
```

- Méthode prompt();

```
prompt('Question ?','chaîne par défaut');
```

# LES OBJETS

## Variables constituées d'une combinaison 'chaine', 'valeur'

- Déclaration

```
var monObjet = {
    'nom'      : 'Tintin',
    'animal'   : 'Milou'
}
```

- Utilisation

```
monObjet.nom            // Tintin
```

# LES FONCTIONS

## Définir du code dans un bloc exécutable

- Déclaration

```
function test(){ ... }
```

- Passage de paramètres

```
function test2(param1, param2){
  return param1 + param2;
}

test2(1,5);       // 6
```

- Fonction auto-exécutée (JQuery)

```
(function(){
   alert('Hello world');
})();
```

24

Utiliser les sélecteurs
Manipuler le HTML
Manipuler les CSS
Manipuler les éléments

# JQUERY : LES BASES

LCProd.net
Grandadam Emmanuel

# UTILISER LES SÉLECTEURS



## La fonction jQuery() ou $()

- Fonction de base pour JQuery

*Exemple :*

```
$("#titre")   => Sélectionne la balise ayant pour ID "titre"
$("h1")       => Sélectionne les balises <h1>
$(".gras")    => Sélectionne les balises ayant pour classe "gras"
$("a,h2")     => Sélectionne les balises <a> et <h2>
$("#")        => Sélectionne toutes les balises
```

*Spécificités JQuery :*

```
:hidden, :visible, :not(s), :first, :last, :header...

$(".annonce:hidden") => Elément de classe annonce qui est masqué
$(":not(img)")       => Ce qui n'est pas une image
```

# MANIPULER LE HTML



## Le texte :

- Méthode text()

  *Les caractères HTML sont convertis*

```
$("#titre).text();                  => Charge le texte
$("#titre).text("bonjour");         => Modifie le texte
```

- Méthode html()

  *Le HTML brut est utilisé*

```
$("#titre).html();
$("#titre).html("<b>bonjour</b>");
```

- Méthode replaceWith()

  *Remplace le contenu d'une balise*

- Méthodes prepend() et append()

  *Ajoute avant ou après du contenu*

- Méthode wrap()

  *Enveloppe avec des balises Ex. : $("#titre").wrap("");*

**Exemple 1.1 / fichier 1.1.html**

# MANIPULER LES CSS



## Les styles :

- Méthode CSS()

  *Permet de lire ou attribuer un style*

- Méthode addClass()

  *Ajoute dynamiquement une classe à un élément*

- Méthodes removeClass()

  *Retire une classe*

- Méthode toggleClass()

  *Alterne l'utilisation d'une classe*

## Les dimensions et position

- Méthodes width() et height()

  *Récupère ou fixe la taille d'un élément*

- Méthode position()

  *Récupère la position d'un élément*

**Exemple 1.2 / fichier 1.2.html**

# MANIPULER LES ÉLÉMENTS



## Ajouter un élément

- Utiliser les méthodes texte() ou html()

- Méthode clone()

  *Permet de dupliquer un élément*

## Supprimer un élément

- Méthode remove()

  *Supprime l'élément du DOM*

- Méthode detach()

  *Supprime l'élément mais permet de le ré-insérer plus tard (variables)*

**Exemple 1.3 / fichier 1.3.html**

29

Utiliser les écouteurs
Utiliser l'élément de l'écouteur
Déclencher un événement
Transmettre l'événement
Les particularités de JQuery

# JQUERY : LES ÉVÉNEMENTS

LCProd.net
Grandadam Emmanuel

# UTILISER LES ÉCOUTEURS



## Ecouteurs

- Connecter des événements aux éléments

```
element.addEventListener('evenement',function(){ ... });
element.evenement(function(){ ... });
```

- Méthode bind() et unbind()

*Permet d'attacher un événement à un élément*

```
$('#lien').bind('click', function() { ... });
```

- Méthodes nommées

```
click, focus, load, mouseover, change, select, submit...

$('#lien').click(function() { ... });
```

# UTILISER L'ÉLÉMENT

## Utiliser l'élément qui à déclenché l'événement

*"L'élément qui a déclenché est intégré à la fonction de l'événement sous le paramètre 'e'"*

- Méthode preventDefault()

  *Annule l'action par défaut*

- Méthode stopPropagation()

  *Annuler la propagation de l'événement -> Paramètre 'e' dans la fonction*

```
$('#lien').click(function(e) {
    e.preventDefault();
    e.stopPropagation();
});
```

**Exemple 2.1 / fichier 2.1.html**

# DÉCLENCHER UN ÉVÉNEMENT



## Déclencher un événement à partir du code

- Méthode trigger()

```
$("#test").trigger("click");
```

Exemple 2.1 / fichier 2.1.html

# TRANSMETTRE L'ÉVÉNEMENT

## Transmettre l'événement aux éléments futur

- Méthode live()

  *Permet de transmettre l'événement aux prochains éléments*

```
$('#liste li').live('click', function(e) { ... });
```

- Méthode delegate()

  *Permet de transmettre l'événement aux prochains éléments spécifiques (li)*

```
$('#liste').delegate('li', 'click', function(e) { ... });
```

# PARTICULARITÉS JQUERY

## Utiliser les événements particuliers

- Evénement hover()

  *Avec toogleClass permet d'alterner au survol*

- Evénement toggle()

  *Prend en charge le va et viens*

```
$('#ligne').toggle(function() { etat1 },function() { etat2 });
```

Utiliser un effet (I/II)
Utiliser les animations
Gérer les enchaînements
Etudes de cas

# JQUERY : LES EFFETS

LCProd.net
Grandadam Emmanuel

# UTILISER UN EFFET

## Fonction d'effets intégrées à JQuery

- Méthode show()
- Méthode hide()
- Méthode fadeIn()
- Méthode fadeOut()
- Méthode slideDown()
- Méthode slideUp()
- Méthode slideToggle()

```
$('h1').show();
```

**Exemple 3.1 / fichier 3.1.html**

# UTILISER UN EFFET

## Modifier la durée des effets

- Permet d'intervenir sur la durée de réalisation de l'effet (en millisecondes) :

  *La durée par défaut est de 300ms*

```
$('h1').fadeIn(300);
$('h1').fadeOut('slow');
```

- Les vitesses d'animation de jQuery

  *Des valeurs par défaut sont utilisables :*

```
slow = 600 ms
fast = 200 ms
```

- Création de vitesse personnalisées :

```
jQuery.fx.speeds.tortue = 2000;
```

# UTILISER LES ANIMATIONS

## Appliquer des effets avancés

- Méthode animate()

```
$('#logo').animate({ modifications }, durée, fonction_fin);
```

*Modifications :*

**Attributs CSS**

*Durée :*

**En millisecondes**

*fonction_fin :*

**Fonction appelée en fin d'animation**

```
$('#logo').animate({
   left: "+=50"
}, 600, function(){
   alert('terminé !');
});
```

Exemple 3.2 / fichier 3.2.html

# UTILISER LES ANIMATIONS

## Utiliser des transitions

- Avec le plugin jQuery Easing Plugin

http://gsgd.co.uk/sandbox/jquery/easing/

```
$('#logo2').animate({
    'padding-left': "+=250","easeOutElastic"]
}, 2000, function(){
    alert('terminé !');
});
```

- Les effets peuvent être enchainés

```
$('#logo').show(300).delay(1000).hide(300);
```

# ETUDES DE CAS

## Gestion des animations/déplacement de la fenêtre du navigateur

- Utiliser la détection de scroll : Méthodes scroll() et scrollTop()

```
$(window).scroll(function(){
   alert( $(this).scrollTop() );
})
```

- Animer la remonter de la page

```
$('body,html').animate({
   scrollTop: 0
}, 800);
```

Exemple 3.3 / fichier 3.3.html

41

AJAX
Manipuler du XML
Manipuler des données JSON
Passer par un formulaire
Utiliser les événement AJAX

# JQUERY : LES ÉCHANGES

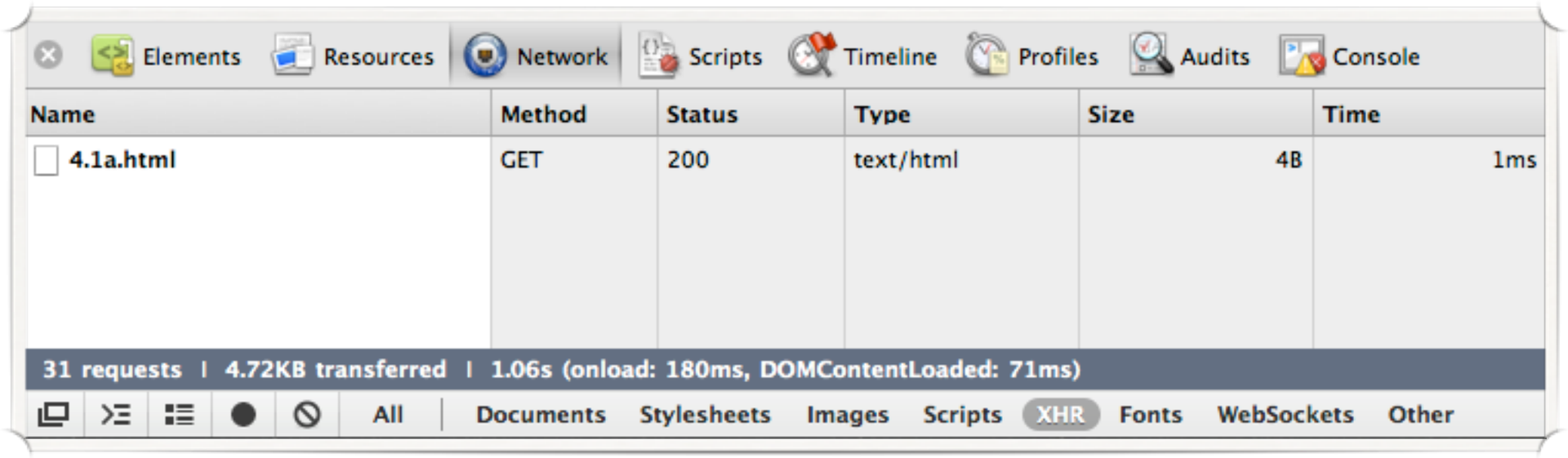
LCProd.net
Grandadam Emmanuel

# AJAX

## Debugage

- Pour Chrome/Safari

*Utiliser le panneau 'Outils de développements' puis*
*-> Onglet 'Network'*
*-> XHR*



# AJAX

## Requête asynchrone

- Méthode GET

```
$.get('fichier.php', function(reponse) { ... });
```

- Méthode POST

```
$.post('fichier.php', function(reponse) { ... });
```

- Méthode AJAX

```
$.ajax({url:'fichier.php',type:'GET',success:function(r){...}});
```

- Méthode LOAD

```
$('#cible').load('fichier.html');
```

Exemple 4.1 / fichier 4.1.html

# AJAX

## Envoi de données

- Soit :

  *Données sous forme d'un Objet {id:12, nom='tintin'}*
  *Données sous forme d'une chaîne "id=12&nom=tintin"*

```
$.ajax({
   url:'fichier.html',
   type:'GET',
   data: {id:12},
   success:function(r){...}
});
```

*URL (en GET)*

```
http://...../fichier.html?id=12
```

**Exemple 4.2 / fichier 4.2.html**

# MANIPULER DU XML



## Extensible Markup Language

- Format de données pour les échanges

```
<adresses>
   <fiche nom='Tintin' />
   <fiche nom='Milou' />
</adresses>
```

- Ouvrir le résultat d'une requête retournant du XML

*Utilisation d'une boucle (each) pour parcourir le contenu*

```
$(xml).find('noeud').each(function(){
   $(this).attr('champ');
}
```

**Exemple 4.3 / fichier 4.3.html**

# MANIPULER JSON

## JavaScript Object Notation

- Format de données pour les échanges

```
{
    "Tintin":4,
    "Milou":2,
    "Castafiore":1,
    "Haddock":3,
    "Tournesol":5
}
```

- Ouvrir le résultat d'une requête retournant du JSON

*Utilisation d'une boucle (each) pour parcourir le contenu*

```
$.each(json, function(key,value){
   alert(key + ':' + value);
})
```

**Exemple 4.4 / fichier 4.4.html**

# PASSER PAR UN FORMULAIRE

## Assemblage des informations du formulaire

- Méthode serialize()

```
$('#monformulaire').serialize();
```

*Donne en GET :*

```
?champ1=valeur1&champ2=valeur2&champ3=valeur3 ...
```

Exemple 4.5 / fichier 4.5.html

# LES ÉVÉNEMENT AJAX

## Jouer sur l'affichage d'un indicateur

- Méthode ajaxStart() et ajaxStop()

```
$('#loader').ajaxStart(function() {
      $(this).show();
})
```

```
$('#loader').ajaxStop(function() {
      $(this).hide();
})
```

Exemple 4.6 / fichier 4.6.html

Utiliser JQuery UI
JQuery UI : Accordéons
JQuery UI : Dialogues
JQuery UI : Onglets

# UTILISER LES PLUG-INS

LCProd.net
Grandadam Emmanuel

# UTILISER JQUERY UI



## Composants

**http://jqueryui.com/download**

- UI Core — Moteur de JQueryUI (demande JQuery)
- Interactions — Outils pour les interactions utilisateur
- Widgets — Amélioration pour les interfaces (dialogues)
- Effets — Effets d'animations avancés

## Thèmes

**http://jqueryui.com/themeroller/**

*Gestion de thèmes CSS pour le kit UI*


# JQUERY UI : ACCORDÉONS

## Créer des sections en accordéons

- Structure HTML

```
<div id="accordion">
    <h4>Section 1</h4>
    <div>Lorem ipsum dolor sit amet</div>
    <h4>Section 2</h4>
    <div>Totam rem aperiam, eaque ipsa quae</div>
</div>
```

- Code Javascript

```
$( "#accordion" ).accordion();
```

Exemple 5.1 / fichier 5.1.html

# JQUERY UI : DIALOGUES

## Créer des boîtes de dialogues

- Structure HTML

```
<div id="dialog" title="Lorem ipsum dolor sit amet">
  <p>Lorem ipsum dolor sit amet</p>
</div>
```

- Code Javascript

```
$('#dialog').dialog();
```

Exemple 5.2 / fichier 5.2.html

# JQUERY UI : ONGLETS

## Créer des boîtes à onglets

- Structure HTML

```
<div id="tabs">
  <ul>
    <li><a href="#tabs-1">Onglet 1</a></li>
    <li><a href="#tabs-2">Onglet 2</a></li>
  </ul>
  <div id="tabs-1">Contenu 1...</div>
  <div id="tabs-2">Contenu 2...</div>
</div>
```

- Code Javascript

```
$( "#tabs" ).tabs();
```

Exemple 5.2 / fichier 5.2.html

Fonction personnalisée pour JQuery
Evénement personnalisé pour JQuery
Créer un widget JQuery

# PLUS LOIN AVEC JQUERY

LCProd.net
Grandadam Emmanuel

# FONCTION PERSONNALISÉE



## Ajoute une fonction dans l'ensemble des fonctions de JQuery

- Utilisation de $.fn

```
$.fn.mafonction = function(){ ... }
```

- Appel de la fonction

```
$("#bouton").mafonction();
```

Exemple 6.1 / fichier 6.1.html

# EVÉNEMENTS CUSTOM

## Créer ses événement personnalisés

- Déclarer un événement avec une fonction

```
$(document).bind('monEvent',function(e, arg1, arg2){
        console.log(arg1);
        console.log(arg2);
    }
);
```

- Appeler l'événement personnalisé (avec paramètres)

```
$(document).trigger('monEvent', [ 'bim', 'baz' ]);
```

# CRÉER UN WIDGET JQUERY



## Définition d'un widget

- Création du widget

```
$.widget('ui.votes', { ... })
```

- Gestion des options

```
options: {cache: true, value: 0}
```

- Constructeur

```
_create: function() { ... }
```

- Méthode publique

```
value: function() {
   return this.options.value;
}
```

Exemple 6.3 / fichier 6.3.html

# LICENCE